#include <iostream>

#include <string>

// Base class Animal

class Animal {

protected:

std::string name;

std::string type;

public:

// Constructor to initialize name and type

Animal(const std::string& animalName, const std::string& animalType)

: name(animalName), type(animalType) {}

// Member function to display basic information

void displayInfo() const {

std::cout << "Name: " << name << std::endl;

std::cout << "Type: " << type << std::endl;

}

};

// Derived class Herbivore

class Herbivore : public Animal {

private:

std::string herbivoreFeature;

public:

// Constructor to initialize herbivoreFeature

Herbivore(const std::string& animalName, const std::string& feature)

: Animal(animalName, "Herbivore"), herbivoreFeature(feature) {}

// Member function to display herbivore-specific information

void displayHerbivoreInfo() const {

displayInfo();

std::cout << "Herbivore Feature: " << herbivoreFeature << std::endl;

}

};

// Derived class Carnivore

class Carnivore : public Animal {

private:

std::string carnivoreFeature;

public:

// Constructor to initialize carnivoreFeature

Carnivore(const std::string& animalName, const std::string& feature)

: Animal(animalName, "Carnivore"), carnivoreFeature(feature) {}

// Member function to display carnivore-specific information

void displayCarnivoreInfo() const {

displayInfo();

std::cout << "Carnivore Feature: " << carnivoreFeature << std::endl;

}

};

int main() {

// Example usage

Herbivore herbivore("Elephant", "Large Ears");

Carnivore carnivore("Lion", "Sharp Claws");

// Display information for herbivore

std::cout << "Herbivore Information:" << std::endl;

herbivore.displayHerbivoreInfo();

std::cout << std::endl;

// Display information for carnivore

std::cout << "Carnivore Information:" << std::endl;

carnivore.displayCarnivoreInfo();

return 0;

}